UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/662,293	09/16/2003	Michel Doyon	10442-30US	9796

20988 7590 01/03/2011

OGILVY RENAULT LLP
1, Place Ville Marie
SUITE 2500
MONTREAL, QC H3B 1R1
CANADA

EXAMINER	
VERDI, KIMBLEANN C	
ART UNIT	PAPER NUMBER
2194	

MAIL DATE	DELIVERY MODE
01/03/2011	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/662,293	DOYON ET AL.	
	Examiner	Art Unit	
	KimbleAnn Verdi	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 October 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-14* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-14* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 – 14 are pending in the application.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. **Claims 1-2 are rejected under 35 U.S.C. 102(e) as being anticipated by Baertsch et al. (hereinafter Baertsch, previously cited) (U.S. Patent 6,470,071 B1).**

4. **As to claim 1**, Baertsch teaches in a computer system, a method for providing improved real time command execution in a non real time operating system, comprising:

executing at least one application at user level mode of said computer system (***i.e. "excel user interface 339, Figure 16, For communication with software system 328, instructions are prepared in excel user interface 339, and then translated by translator 331 before being received by Perl script unit 333", col. 15, lines 25-28***);

having said at least one application (***i.e. "excel user interface 339", Figure 16***) at said user mode level (***i.e. -excel is a user interface which executes at user mode as shown in Figure 71-, "As illustrated, interface 730 includes a plurality of user interfaces 732, which interfaces with operating system kernel 734", col. 72, lines 54-56***) determine a sequence to be followed for a set of commands (***i.e. "frame sequence 310"- is specified by user using excel-,"An exact sequence of image frames and associated acquisition parameters is needed in advance for a particular image acquisition. Accordingly, one can specify, for each frame, the readout delay relative to x-ray pulse, the detector parameters, etc. A description of such attributes is captured in a frame sequence 310 of script 309. Program applications configure the data acquisition system through the frame sequence structure and then trigger the system to initiate acquisition of the frames", col. 14, lines 10-18, "Referring to FIG. 16, the event compiler 408 takes a Perl script as its input. Data from an Excel user interface 339 can alternatively be used to generate the Perl script with translator 331", col. 77, lines 22-25, "FIG. 15 is a block diagram showing the flow of control information and data through system 300 during image acquisition. As illustrated, frame sequence 310 is created by way of script 309", col. 14, lines 39-42)***;

providing (***i.e. creating and sending***) from said at least one application ***(i.e. - using "Excel user interface 339", Figure 14 to create the script***) said sequence of commands (***i.e. "FIG. 15 is a block diagram showing the flow of control information and data through system 300 during image acquisition. As***

illustrated, frame sequence 310 is created by way of script 309", col. 14, lines 39-42) to a privileged mode (***i.e. DFN device driver 314 operates at kernel mode, Figure 71***) of said computer system (***i.e. "Frame sequence 310 is then translated into event sequence 312 using a compiler, which knows the details of the target control hardware. Event sequence 312 is received by test control unit 311, then sent to DFN device driver 314, over computer communication bus 302, and finally to detector framing node 304. The event sequence 312 is then stored in preparation for execution", col. 14, lines 42-48***) to be executed in real time (***i.e. "Once the event sequence 312 is known, the details are transmitted to DFN 304 for execution in real-time", col. 14, lines 36-38***);

storing said sequence of commands (***i.e. "Event sequence 312, Figure 15"***) in a command queue (***i.e. "Event Queue 322, Figure 15, Event sequence 312 is received by test control unit 311, then sent to DFN device driver 314, over computer communication bus 302, and finally to detector framing node 304. The event sequence 312 is then stored in preparation for execution", col. 14, lines 44-48, "As illustrated, detector framing node 304 communicates commands and responses with computer communication bus 302 by way of acquisition control unit 324. Event sequence 312 is communicated to event queue 322 by way of acquisition control unit 324, col. 14, lines 61-65***) to be accessible (***i.e. "event sequence initiated"***) from a privileged mode level (***i.e. "kernel, Begin Sequence command sent over computer communication bus 302"***) of said computing system (***i.e. "Event sequence 312 is initiated by sending a Begin Sequence command***

over computer communication bus 302. The extent of real-time control allotted to host computer 114 is confined to a determination of when event sequence 312 will begin", col. 14, lines 48-53, DFN device driver 314 operates at kernel mode, Figure 71, - event sequence stored in event queue 322 is accessible to the host from privileged mode level of the kernel when the host initiates event sequence using the Begin Sequence Command which is sent to the DFN 304 via Device driver 334 to initiate the event sequence stored in event queue 322 of DFN 304- "Device driver 334 is a kernel-mode program that provides an interface to access hardware and also controls DFN hardware interactions with the operating system", col. 72, lines 51-53); and

executing one at a time each of said commands (***i.e. "event instructions"***) from said stored sequence of commands (***i.e. "event sequence", "According to an embodiment of the present invention, the instructions are event instructions, known collectively as an event sequence 312. Each event instruction is executed at well-timed intervals. Event instructions trigger events that control external devices, such as through commands communicated over bus interfaces. For example, event instructions include 32 bit control words that are sent over image detection bus 377 to image detection system 112, and x-ray pulse trigger commands sent over real-time bus 379 to radiation generation system 109. Based on frame sequence 310, a complete list of such event instructions to be performed is constructed. The event sequence 312 need not be constructed in real-time and is therefore easily executed on host computer 114 running a non-***

real time operating system to support an event compiler. Once the event sequence 312 is known, the details are transmitted to DFN 304 for execution in real-time", col. 14, lines 22-26).

5. **As to claim 2**, Baertsch teaches wherein a plurality of sequences of asynchronous commands is provided (**i.e.** ***"Frame sequence 310 is then translated into event sequence 312 using a compiler, which knows the details of the target control hardware", col. 14, lines 42-44***), each sequence being related to a corresponding application thread (***i.e. -excel is a user interface which executes at user mode as a user process or thread as shown in Figure 71 and creates the sequence-, "As illustrated, interface 730 includes a plurality of user interfaces 732, which interfaces with operating system kernel 734", col. 72, lines 54-56 "FIG. 15 is a block diagram showing the flow of control information and data through system 300 during image acquisition. As illustrated, frame sequence 310 is created by way of script 309", col. 14, lines 39-42, "Referring to FIG. 16, the event compiler 408 takes a Perl script as its input. Data from an Excel user interface 339 can alternatively be used to generate the Perl script with translator 331", col. 77, lines 22-25***), further wherein said storing of a sequence of commands is performed in a corresponding queue (***i.e. "As illustrated, detector framing node 304 communicates commands and responses with computer communication bus 302 by way of acquisition control unit 324. Event sequence 312 is communicated to event queue 322 by way of acquisition control unit 324, col. 14, lines 61-65***) from

the execution of said corresponding application thread queue (***i.e. "Program applications configure the data acquisition system through the frame sequence structure and then trigger the system to initiate acquisition of the frames", col. 14, lines 15-18***).

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. **Claims 3 – 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Baertsch et al. (hereinafter Baertsch, previously cited) (U.S. Patent 6,470,071 B1) in view of Dingwall et al. (hereinafter Dingwall, previously cited) (U.S. Patent No. 5,903,752).**

8. **As to claim 3**, Baertsch does not explicitly disclose wherein a synchronous command is added to said sequence of commands, said at least one application sleeping until said synchronous command is executed.

9. However Dingwall teaches wherein a synchronous (***i.e. real-time***) command is added to said sequence of commands, said at least one application sleeping (***i.e. application task is asleep (dormant/locked) until interrupted, 818, Fig. 8***) until said synchronous command is executed (***i.e. RT Scheduler 30, Fig. 2, releases scheduling lock which allows real-time tasks to pre-empt the current (asynchronous) process, col. 3, lines 59-61***).

10. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified the invention of Baertsch to incorporate the features of Dingwall. One of ordinary skill in the art would have been motivated to make the combination because this allows real-time programming with support for the presentation of natural data types, without allowing other operations to disrupt the delivery and playback of the audio and video data (***col. 1, lines 66-67 and col. 2, lines 1-3 of Dingwall***).

11. **As to claim 4**, Baertsch does not explicitly disclose wherein a synchronous command is added to said sequence of asynchronous commands, said corresponding application thread sleeping until said synchronous command is executed.

12. **However** Dingwall teaches wherein a synchronous command is added to said sequence of asynchronous commands, said corresponding application thread sleeping (***i.e. application task is asleep (dormant/locked) until interrupted, 818, Fig. 8***) until said synchronous command is executed (***i.e. RT Scheduler 30, Fig. 2, releases***

scheduling lock which allows real-time tasks to pre-empt the current (asynchronous) process, col. 3, lines 59-61).

13. The motivation for further modifying Baertsch with the teachings of Dingwall is the same as provided in the rejection of claim 3 above.

14. **As to claim 5**, Baertsch does not explicitly disclose wherein said non real time operating system is MICROSOFT WINDOWS™ and said storing said sequence of commands is performed through execution of a driver routine from a DLL file.

15. However Dingwall teaches wherein said non real time operating system is MICROSOFT WINDOWS™ (***i.e. environment of WINDOWS™, col. 3, lines 33-34)*** and said storing said sequence of commands is performed through execution of a driver routine from a DLL file ***(Virtual Device Driver (VxD) is dynamic link library (DLL), col. 3, lines 33-36***).

16. The motivation for further modifying Baertsch with the teachings of Dingwall is the same as provided in the rejection of claim 3 above.

17. **As to claim 6**, Baertsch does not explicitly disclose wherein said providing said sequence of commands involves said commands being pushed one at a time into said sequence through system call.

18. However Dingwall teaches wherein said providing sequence of commands involves said commands being pushed one at a time into said sequence through system call (***i.e. interrupt occurs which causes the processor to switch to VxD interrupt mode and execute RT interrupt handler 32, Fig. 2, col. 4, lines 51-23, RT interrupt handler 32, Fig. 2, wake up associated real-time task***).

19. The motivation for further modifying Baertsch with the teachings of Dingwall is the same as provided in the rejection of claim 3 above.

20. **As to claim 7**, Baertsch does not explicitly disclose wherein at least one of said stored commands is a branch command to control the order of execution of said stored commands.

21. However Dingwall teaches wherein at least one of said stored commands is a branch command to control the order of execution of said stored commands (***i.e. RT scheduler 30, Fig. 2, schedules task preemptively by priority and allows interrupt handlers 32, Fig. 2, to make real-time tasks ready for execution without preemption, col. 3, lines 54-62***).

22. The motivation for further modifying Baertsch with the teachings of Dingwall is the same as provided in the rejection of claim 3 above.

23. **As to claim 8**, Baertsch does not explicitly disclose wherein said executing said commands from said stored sequence of commands is done at a different privileged mode level system.

24. However Dingwall teaches wherein said executing said commands from said stored sequence of commands is done at a different privileged mode level system (***i.e. Virtual Device Driver (VxD), 28, Fig. 2, run at most privileged level col. 3, lines 36-37***).

25. The motivation for further modifying Baertsch with the teachings of Dingwall is the same as provided in the rejection of claim 3 above.

26. **As to claim 9**, Baertsch does not explicitly disclose wherein said different privileged mode level is that of Interrupt Service Routine, whereby delay between the execution of successive commands is minimized.

27. However Dingwall teaches wherein said different privileged mode level is that of Interrupt Service Routine (***i.e. Virtual Device Driver (VxD), 28, Fig. 2, which is interrupt driven, runs at most privileged level col. 3, lines 36-38***), whereby delay between the execution of successive commands is minimized (***i.e. improves real-time response col. 2, line 49-50***).

28. The motivation for further modifying Baertsch with the teachings of Dingwall is the same as provided in the rejection of claim 3 above.

29. **As to claim 10**, Baertsch does not explicitly disclose wherein said non real-time operating system is MICROSOFT WINDOWS™.

30. However Dingwall teaches wherein said non real-time operating system is MICROSOFT WINDOWS™ (***i.e. environment of WINDOWS™, col. 3, lines 33-34***).

31. The motivation for further modifying Baertsch with the teachings of Dingwall is the same as provided in the rejection of claim 3 above.

32. **As to claim 11**, Baertsch does not explicitly disclose wherein said sequence of commands process a same data set.

33. However Dingwall teaches wherein said sequence of commands process a same data set (***i.e. task needs to process data in buffer stored by audio/video device, col. 4, lines 59-60***).

34. The motivation for further modifying Baertsch with the teachings of Dingwall is the same as provided in the rejection of claim 3 above.

35. **As to claim 12**, Baertsch does not explicitly disclose wherein said same data set is a video camera image being captured and processed in real-time.

36. However Dingwall teaches wherein said same data set is a video camera image being captured and processed in real-time (***i.e. task needs to process data in buffer stored by audio/video device, col. 4, lines 59-60***)(***i.e. example task used to perform capture or playback of audio/video, col. 4, lines 5-6***).

37. The motivation for further modifying Baertsch with the teachings of Dingwall is the same as provided in the rejection of claim 3 above.

38. **As to claim 13**, Baertsch does not explicitly disclose wherein said providing said sequence of commands involves said commands being pushed one at a time into said sequence through a system call.

39. However Dingwall teaches wherein said providing said sequence of commands involves said commands being pushed one at a time into said sequence through a system call (***i.e. interrupt occurs which causes the processor to switch to VxD interrupt mode and execute RT interrupt handler 32, Fig. 2, col. 4, lines 51-23, RT interrupt handler 32, Fig. 2, wake up associated real-time task***).

40. The motivation for further modifying Baertsch with the teachings of Dingwall is the same as provided in the rejection of claim 3 above.

41. **As to claim 14**, Baertsch does not explicitly disclose wherein said storing said sequence of commands is performed through execution of a driver routine from a system file.

42. However Dingwall teaches wherein said storing said sequence of commands is performed through execution of a driver routine (***i.e. Virtual Device Driver***) from a system file (***i.e. Virtual Device Driver (VxD) is dynamic link library (DLL), col. 3, lines 33-36***).

43. The motivation for further modifying Baertsch with the teachings of Dingwall is the same as provided in the rejection of claim 3 above.

Response to Arguments

44. Applicant's arguments filed on October 26, 2010 have been fully considered but they are not persuasive. In response to the Non-Final Office Action dated July 27, 2010, applicant argues in regards to claims 1-14:

(1) ***The Examiner's allegation that Baertsch teaches "storing said sequence of commands in a command queue to be accessible from a***

privileged mode level of said computer system" is incorrect (page 2, lines 11-13).

In response to argument (1), examiner respectfully disagrees and notes that Baertsch discloses storing said sequence of commands in a command queue to be accessible from a privileged mode level of said computer system. Baertsch teaches ***"Event sequence 312 is received by test control unit 311, then sent to DFN device driver 314, over computer communication bus 302, and finally to detector framing node 304. The event sequence 312 is then stored in preparation for execution", col. 14, lines 44-48, "As illustrated, detector framing node 304 communicates commands and responses with computer communication bus 302 by way of acquisition control unit 324. Event sequence 312 is communicated to event queue 322 by way of acquisition control unit 324, col. 14, lines 61-65***, which represents storing said sequence of commands in a command queue***. "Event sequence 312 is initiated by sending a Begin Sequence command over computer communication bus 302. The extent of real-time control allotted to host computer 114 is confined to a determination of when event sequence 312 will begin", col. 14, lines 48-53, DFN device driver 314 operates at kernel mode, Figure 71,*** which represents to be accessible from a privileged mode of said computer system since event sequence stored in event queue 322 is accessible to the host from a privileged mode level of the kernel when the host initiates the event sequence using the Begin Sequence Command. In addition the Begin Sequence command is sent

from the host to the DFN 304 via Device driver 334 to initiate the event sequence stored in event queue 322 of DFN 304, which represents privileged mode of the computer system since ***"Device driver 334 is a kernel-mode program that provides an interface to access hardware and also controls DFN hardware interactions with the operating system", col. 72, lines 51-5.***

Conclusion

45. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

46. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

47. Any inquiry concerning this communication or earlier communications from the examiner should be directed to KimbleAnn Verdi whose telephone number is (571)270-1654. The examiner can normally be reached on Monday-Friday 7:30am-5:00pm EST..

48. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sough can be reached on (571)272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

49. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Hyung S. SOUGH/
Supervisory Patent Examiner, Art Unit 2194
01/02/11
KV
December 30, 2010